

AN2016.02

Using the Prio API for wired streaming with a PrioVR Hub

Nick Leyder

Lead Software Engineer, Yost Labs

INTRODUCTION

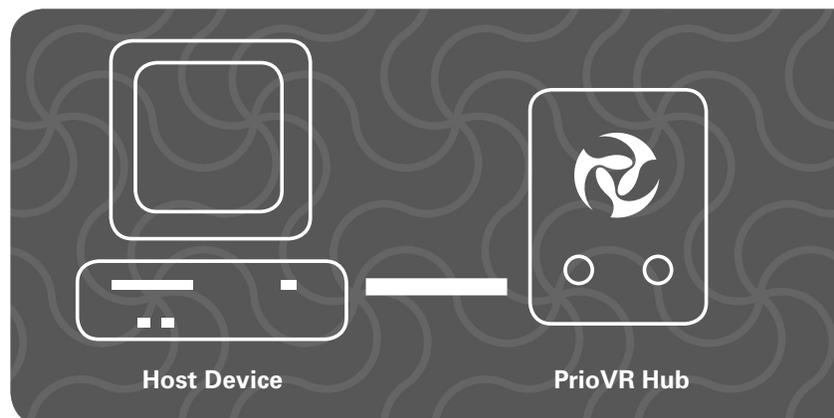
The PrioVR Suit is a full-body motion tracking suit that uses 19 PrioVR Sensors, and a PrioVR Hub to do so. The PrioVR Sensors are inertial measurement units that give accurate drift-free absolute orientations of the user's body. The PrioVR Hub collects the data from the PrioVR Sensors, and sends it over USB to the computer. This is shown in Figure 1.

The purpose of this document is to provide an overview of streaming data from a PrioVR Suit using the PrioVR Hub. The example code below illustrates the use of the Prio API to read the global orientations from the PrioVR Suit and print them to the screen.

DESCRIPTION

In order to achieve wired streaming, the Prio API must:

1. Find the PrioVR Hub connected to the computer
2. Start streaming the data
3. Read the data
4. Stop streaming and deinitialization the Prio API to allow for later use.



Step 1: Creating the PrioVR Hub

The first step in using a PrioVR Suit through the YOST Prio API is to create a PrioVR Hub to work with.

```
// An unique identifier used by the PrioVR API to identify the PrioVR Hub
prio_device_id hub_device;
// The communication port of a PrioVR Hub
prio_ComPort port;
// An index into the array of communication ports found by the Prio API
uint8_t com_offset = 0;

// Find all of the communication ports that corresponded to PrioVR Hub
prio_findPorts(PRIO_HUB);
// Get the communication port at the index of com_offset
prio_getPort(&port.port_name, com_offset , &port.device_type);
// Create a PrioVR Hubobject from the communication port
prio_createHub(port.port_name.c_str(), &hub_device);
```

This code will find a Hub connected to the host PC at the given offset in the communication ports (com_offset) If none is present, bs_device will be set to an error value. This ID will be passed to further functions that reference the Hub .

Step 2: Start Streaming

With the PrioVR Hub created, it is now time to instruct the PrioVR Hub on what data to gather and to start gathering that data from the PrioVR suit.

```
// This starts streaming untared orientations, at an interval of 10000 mircoseconds
// forever
prio_hub_startStreamingWired(bs_device, PRIO_STREAM_UNTARED_ORIENTATION_AS_QUATERNION, 10000,
PRIO_STREAM_DURATION_INFINITE);
```

This code will start streaming untared quaternions, the other options are:

1. PRIO_STREAM_TARED_ORIENTATION_AS_QUATERNION
2. PRIO_STREAM_ALL_CORRECTED_COMPONENT_SENSOR_DATA
3. PRIO_STREAM_CORRECTED_GYRO_RATE
4. PRIO_STREAM_CORRECTED_ACCELEROMETER_VECTOR
5. PRIO_STREAM_CORRECTED_MAGNETOMETER_VECTOR
6. PRIO_STREAM_ALL_RAW_COMPONENT_SENSOR_DATA
7. PRIO_STREAM_RAW_GYROSCOPE_RATE
8. PRIO_STREAM_RAW_ACCELEROMETER_DATA
9. PRIO_STREAM_RAW_MAGNETOMETER_DATA

Up to 512 bits in any combination of up to 8 options can be streamed simultaneously.

Step 4: Read the Data

At this point, orientation data will be automatically obtained from the PrioVR Hub at a regular interval. All that remains is to read the data.

```
while(true) // For as long as you need suit data for
{
// The header data from the packet
prio_StreamHeaderData header_data;
// The packer data from the packet, sized for 19 sensors
U8 packet_data[304];
// Retrieve the latest packet from the PrioVR Hub prio_hub_getLastStreamingPacket(hub_device,
&header_data, &packet_data[0]);
// A float array for storing 19 quaternions
float packet_quats[76];
// Copy the packet data into the float array
memcpy(packet_quats, packet_data, 304);
int i=0;
do
{
// Printing out the sensor quaternions
printf("( %f, %f, %f, %f)", packet_quats[i], packet_quats[i+1], packet_quats[i+2], packet_quats[i+3]);
// Increment the quaternion array by the 4 consumed data
i += 4;
} while( i < 76) // Loop until all of the packet quaternions have been printed
}
```

The “prio_StreamHeaderData” structure used here is a container for two uint8s that store the battery level, the state of the PrioVR Hub buttons, and two prio_Joystick that hold the joystick data. The “prio_JoystickData” structure is a container for four uint8s that hold the x_axis, y_axis, trigger state, and button state.

The “packet_data” used here is a array that holds all of the data streamed by the PrioVR Hub for 19 sensors. This data is in an array where a sensors quaternion data is 16 spots in the array ([0-15],[16-31]...[287-303]). The code then converts that data into a float array for ease of use, and prints the data to the user. A quaternion can be converted to any other major orientation format, including rotation matrix, axis-angle, or Euler angles.

Step 5: Finishing Up

Before the program ends there are a few things that need done, those are stopping streaming, and shutdown the Prio API.

```
// Stops the streaming of the PrioVR Hub
prio_hub_stopStreamingWired(hub_device);
// Deinitialize the Prio API
prio_deinitAPI();
```

Complete Example

```
#include <stdio.h>
#include <time.h>
#include "prio_api_export.h"

#define PRIO_STREAM_DURATION_INFINITE 0xffffffff

void main()
{
    // An unique identifier used by the PrioVR API to identify the PrioVR Hub
    prio_device_id hub_device;
    // The communication port of a PrioVR Hub
    prio_ComPort port;
    // Allocate an char array of 64 chars for the port name
    port.port_name = new char[64];
    // An index into the array of communication ports found by the Prio API
    uint8_t com_offset = 0;

    // Find all of the communication ports that correspond to PrioVR Hub
    prio_findPorts(PRIO_HUB);
    // Get the communication port at the index of com_offset
    prio_getPort(port.port_name, com_offset, &port.device_type);
    // Create the Hub
    prio_createHub(port.port_name, &hub_device);

    // Start streaming data from the Prio suit
    prio_hub_startStreamingWired(hub_device, PRIO_STREAM_UNTARED_ORIENTATION_AS_QUATERNION, 0,
    PRIO_STREAM_DURATION_INFINITE);

    while(true) // For as long as you need suit data for
    {
        // The header data from the packet
        prio_StreamHeaderData header_data;
        // The packer data from the packet, sized for 19 sensors
        U8 packet_data[304];

        // Retrieve the latest packet from the PrioVR Hub
        prio_hub_getLastStreamingPacket(hub_device, &header_data, &packet_data[0]);

        // A float array for storing 19 quaternions
        float packet_quats[76];
```

continued . . .

```
// Copy the packet data into the float array
memcpy(packet_quats, packet_data, 304);

int i=0;
do
{
// Printing out the sensor quaternions
printf("( %f,%f,%f,%f)",packet_quats[i],packet_quats[i+1],packet_quats[i+2],packet_quats[i+3]);

// Increment the quaternion array by the 4 consumed data
i += 4;
} while( i < 76) // Loop until all of the packet quaternions have been printed
}

// Stops the streaming of the PrioVR Hub
prio_hub_stopStreamingWired(hub_device);

// Deinitialize the Prio API
prio_deinitAPI();
}
```



Made in USA. Patents: 8498827,
8682610, 9255799, 9354058.
Additional patents pending.

About Yost Labs, Inc. We are a fast growing private company based in historic Portsmouth, Ohio. With over a decade of experience in low-latency inertial sensor innovation, we enable motion tracking in many of today's and tomorrow's most exciting products. We make virtual reality interactive. We stabilize drones and navigate autonomous cars. We measure human motion for athletic performance and rehabilitation. We tell you where you are heading and how fast you are getting there. Yost Labs' innovation has been recognized with numerous patents with additional patents pending. Our customers and value-added resellers include the US Navy, US Air Force, NASA, US Army Corps of Engineers and over 1,000 leading technology firms and academic institutions around the world.